Robotic Arm Design, Development and Control for Agriculture Applications

Rajesh Kannan Megalingam, Gedela Vamsy Vivek, Shiva Bandyopadhyay, Mohammed Juned Rahi
Dept. of Electronics and Communication Engineering
Amrita School of Engineering, Amritapuri
Amrita Vishwa Vidyapeetham, Amrita University, India
rajeshkannan@ieee.org, v.vivek.1997@ieee.org, shiva1bandyopadhyay2@gmail.com, junedrahi@gmail.com

***Abstract*—In this work we analyze various methods and techniques for the design and development of robotic arm for agricultural applications. In this paper, a technique by which a robotic arm is controlled using a depth sensor is proposed. The depth sensor captures the user input and inverse kinematics algorithm is used to define the motion of the robotic arm. We develop a robotic arm which imitates the action of human arm. The movement of the human arm in 3D space is captured, processed and replicated by the robotic arm using inverse kinematics. In first place a Kinect sensor captures the input and coordinates of the shoulder, elbow and wrist joints of the user are obtained from it. The angles at each joint required to completely describe the position of the robot's joints are calculated using an inverse kinematics technique as mentioned above. Through a proper microcontroller board, the robotic arm end effecter moves to the required position which can be used to pluck a fruit or prune a branch etc.**

***Index Terms*—Robotic arm, inverse kinematics, Kinect sensor, end effecter.** *(key words)*

I. Introduction

Agricultural sector is facing various challenges which are social including shortage in farmland, land ownership; economic challenges including globalization, shortage in labor etc.; and environmental challenges including climate changes, food security etc. Agricultural production is expected to double, to meet the food demands, as the world's population is expected to reach 9 billion by that time. Technological intervention in agricultural sector can be of great help in addressing this challenge. In this paper, we present a novel approach to address the problem of harvesting in agricultural sector. This could be harvesting of apples or oranges in orchards, pruning trees on highways or households, harvesting coconuts etc. A startup in Menlo Park, California, USA called Abundant Robotics Inc. is trying to automate the harvesting of apples with agricultural robotics [10]. University of Auckland, New Zealand, is developing a Multipurpose Orchard Robot to deliver pollination and harvesting services [11].

We propose a Kinect based robotic arm control for addressing issues related to harvesting. A Kinect is a device that can sense the gestures of a human being, can process these signals and can be used to control any robot. Kinect is a product of Microsoft for Xbox 360 which was a revolutionary change in the field of gaming. One just needs to move his body and limbs so that Kinect gets its input and the same can be replicated in the game without the need for joystick or a key board. But in case of Kinect, the user actually gets the feeling of real time simulation in the game without the need to touch and use any handheld device. A Kinect device can receive the input from the user as body or hand gestures. These input signals would be processed by a host processor and sent to the embedded systems based controller being used which would in turn send signals to the robotic arm using Bluetooth wireless interface. By this way the robotic arm embedded controller can control the arm remotely. This research work concentrates on agricultural application of robotic arms using hand gestures to cut fruits, branches or twigs. Positioning and control of the robotic arm can be made simpler and training personnel to control the robotic arm becomes much easier.

The main concept this paper deals with is the inverse kinematics which is comparatively complex unlike forward kinematics. In forward kinematics, the joint angles are input such that the end effector coordinates of the system can be determined whereas in inverse kinematics, the end effector coordinates are the input according to which the joint angles are determined. Forward kinematics is straight forward and excellent when the end effector coordinates need to be calculated only once and the robotic arm is not very complex. But for real time calculation of end effector position which can change dynamically, inverse kinematics is employed though it is though it is computationally expansive and takes long time to arrive at the result. For our proposed work, we chose inverse kinematics as the end effector position need to be calculated dynamically, repeatedly.

II. Motivation

In India 65% of people are dependent on agriculture. Sufficient skilled manpower is not available in many industrial sectors in India including agricultural sector and it affects the growth of the country. Agriculture is the backbone of Indian economy contributing to the overall economic growth of the country. Agriculture contributes to 14% of the overall GDP in India. With the population expected to reach 1.35 billion by 2020, agricultural production must increase if it is to meet the increasing demands for food and bioenergy. Agricultural industry requires a modern and innovative technology to take care of these challenges. Robotics is one of the five technology trends that is driving innovations in the modern day Industrial Revolution. It is estimated that a record 14,232 robots which costs $840 million, were ordered from North American robotic companies in the year 2015[9]. Robotics can play an important

978-1-5090-4559-4/17/$31.00©2017IEEE

role in increasing the efficiency of agriculture productivity given limited land, water and labor resources. They can be employed in various agricultural environments including large-area cultivation, orchards, horticulture, nurseries etc. It can increase the production rate in agriculture sector.

III. Related Works

Numerous methods exist for for manipulating robotic arms - position controlled manipulators, joystick based controllers [1], speech and gesture based controller [2], and sensor based interfaces [3] to name a few. Position controlled manipulators controller are too costly and its uses are mostly for remotely operated underwater vehicles. Other method which is used for manipulating robot arm is speech based, however it cannot be used for complex manipulation such as industry applications where noise level is high. In the research work published in [4] an algorithm for real time gesture recognition is used to control the robot. The robot recognizes a particular set of commands which translates into the robot doing some specific number movements. The ability of this robot is thus limited to directed movement. The work proposed here provides an easier interface to manipulate a robotic arm using a Microsoft Kinect and inverse kinematics. In the research work published in [5] the author uses triangular algorithm for inverse kinematics problem. Jacobian iteration is faster than the triangular algorithm although the latter provides a solution to singular matrix whereas Jacobian iteration fails to do so. In the research work published in [6] the author uses cyclic coordinate descent (CCD) inverse kinematics solutions in multi-joint chains. CCD takes a series of iterations before reaching to a solution and also generate improper joint rotations. CCD has two major drawbacks, firstly the joints close to the end effector rotate more than the joints close to the immobile joint, and the kinematic chain will appear to roll in on itself. Secondly CCD is an iterative method which moves joint in opposite order according to their importance. The movements seem unnatural in case of humanoid robots because outer joints are moved first. The paper [7], gives an idea about a gesture and position of human arm based interaction with robotic arms using wireless accelerometer. Our work is similar in the sense that Kinect control is also gesture based. It involves image processing, whereas the paper [7] uses acceleration due to gravity to determine and recognize gestures. In [17] we have briefly discussed about the Kinect based control of robotic arm.

IV. ANALYSIS OF DIFFERENT METHODS FOR ROBOTIC ARM CONTROL

Different methods for controlling robotic arms. Each method has its own ups and downs. Each method is suitable for a particular group of tasks and hence, applying the same control method in all scenarios in unwise. Robotic arms used in industries repeat the same process again and again. So, they need to be trained only once. The working environment remains same. A welding robot repeats the same welds, i.e., executes the same end effector path over and over again. But, this is not the case with a robotic arm used for harvesting fruits. The working environment is dynamic and unpredictable. In such conditions, a highly accurate automatic robotic arm is very difficult to implement. So at least a semi-automatic human interaction is desirable. Here, the same control mechanism used for industrial robots, like a 3D – joystick, will not be practical because of its complexity. More human friendly techniques such as gesture recognition and motion tracking are preferred here. Owing to the reasons mentioned above, motion tracking stands out to be the preferred method.

A. Accelerometer Based Control

An accelerometer based robotic arm control system may use a single accelerometer or a group of accelerometers depending on the number of DOFs required. A simple and straightforward implementation of this is a 2 – accelerometer based control. This method of control can track the users arm and control two joints of the robotic arm. For example, consider the case where an accelerometer is placed just above the elbow and one just below the elbow. The accelerometers send a mapping of the forces experienced by them in x, y and z directions to a host processor. The host processor applies a suitable gesture and pattern recognition algorithm like HMM or DTW to recognize the gesture. The appropriate command is sent to the robotic arm, which responds to the gesture accordingly by executing a certain set of actions using the on-board controller. A similar implementation exists, where the users arm is tracked and the robotic arm mimics the user. The motion of the user's arm is taken as such and replicated as such and hence pattern and gesture recognition algorithms need not be used here. The method needs the user to continuously control the robotic arm. It is preferred in scenarios where the environment is dynamic as the user can control the arm in real time, reducing errors which are bound to occur in such an environment if the control solely depends on artificial intelligence [13].

B. Flex Resistor Based Control.

Flex resistors are variable resistors. These are bendable and the resistance varies with the amount of bend. A wearable glove is made such that the flex resistors are positioned at the joints of the human arm. Thus, the movement of the user's arm is sensed by the flex sensors. The data is then used by a microcontroller to control the robotic arm [15].

C. Exoskeleton Based Control

Numerous exoskeletons have been developed and tested for various parts of the human body, including hands, arms and legs. The exoskeletons offer many DOFs. The degree of bending of the joints is measured by the exoskeleton. Exoskeletons have been widely used because of their capability to produce tactile feedback, which is very important in cases where the manipulator can be directly seen [14].

D. Reflective Marker System Based Control

This method uses reflective markers placed on different parts of the human arm. Several images of the reflective markers are captured from different angles are analyzed to determine the position of the arm. These systems tend to take a longer processing time because images from all the cameras have to be analyzed to determine the position of the arm. The reflective markers are first identified. Ambiguities may arise in this stage if the markers are placed too close by. So, this method is not preferred in scenarios where fingers have to be traced. Also, as he number of cameras increases, there is a trade – off of process time for increase in accuracy. This method of limb tracking is mainly used to capture human motion for animation [14].

V. System Architecture

The system architecture has 5 blocks as shown in Figure 1. Data from the physical world enters the system through the sensor (Kinect). It is processed by the following three blocks – Portable computer, Kinect SDK, and microcontroller. The processed data is now given as output in the form of motion of the actuators present in the robotic arm. The data is processed in a sequence by the Kinect, computer and microcontroller to achieve the desired end-effector position.

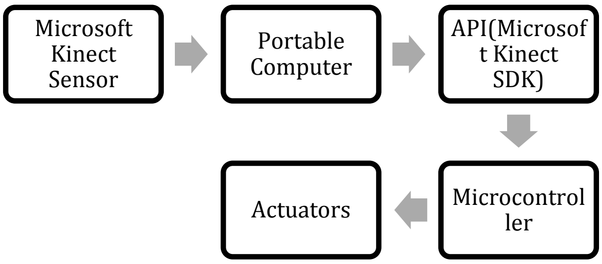

Figure 1. The block diagram of the robotic arm.

A. Microsoft Kinect Sensor

The Microsoft Kinect sensor is shown in Figure 2, with the important parts labeled. Microsoft Kinect sensor features the following parts – Depth sensor (IR laser projector and IR CMOS detector), RGB camera and a microphone array for acoustic purposes. The depth sensor is applied to capture the motion of an object as depth data and skeletal data. Kinect uses Structured – light 3D imaging technique. The IR laser projector projects narrow stripes of IR onto a target 3D object. The straight IR stripe gets deformed and this deformation of the stripe as seen from a point of view different to that of the source is recorded by the IR camera, shown in Figure 3. To get depth data from the received light pattern, the whole frame is compared to a “flat surface” stored in the memory of the Kinect, which was defined earlier during calibration. Thus, distance of each point to the camera is calculated and thus the 3D depth data is generated. The projection of a multiple number of stripes gives a Hi- Resolution depth data. Kinect SDK V1.8 provides inbuilt for joint and skeletal tracking.

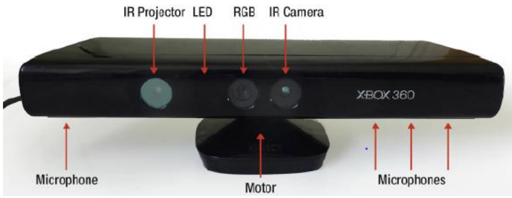

Figure 2. The image of a Kinect.

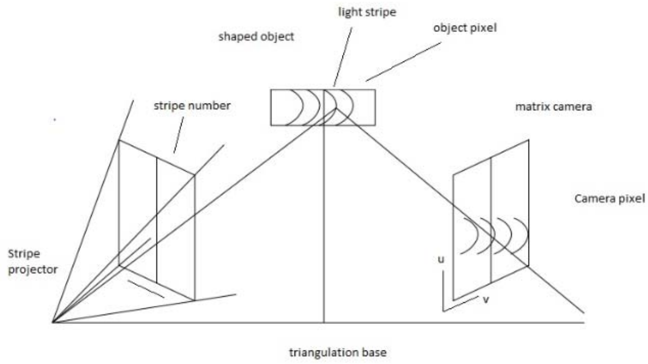

Figure 3. Light falls on 3D object and a camera at a different point of view captures the distorted stripe of light.

B. Portable Computer

The We used a portable computer as the host processor which receives the data from the Kinect via USB. The portable computer has the necessary APIs installed in it for communication with the Kinect. Using the data received from the Kinect the end effector position is calculated by the PC. Next, the joint angles are calculated using an inverse kinematics algorithm, which has been discussed later in this paper. The calculated angles are sent to the microcontroller as ASCII text via serial communication.

C. Microsoft Kinect SDK

Microsoft Kinect SDK V1.8 contains a set of routines and tools essential for specifying how the PC should interact with Kinect. The PC gets the position of the end effecter (wrist). The computer uses this data to calculate the angle of each joint of the arm using Inverse kinematics algorithms. The microcontroller on the arm is given appropriate commands. The microcontroller drives each of the actuators to the required position.

D. Microcontroller

The microcontroller module we used is an Arduino UNO R3. The Arduino UNO receives the angles for each of the joints from the PC as ASCII text. The ASCII text is converted to integer form by using Arduino library functions. These angles are tested for validity, so that they don’t go out of range of the actuator’s capability. The rotation of each of the actuators is calibrated, such that the Arduino is able to generate an appropriate 8-bit PWM signal to move the actuator to the required position. The key features of the microcontroller module are given in Table 1:

Table 1. Key properties of microcontroller board

Part	Specification
Microcontroller	Atmel ATmega 328p
Clock Speed	16MHz
Memory	32KB flash, 2KB RAM

E. Actuators

We built our own 3 DOF robotic arm which has similar joints as compared to human hand. The robotic arm was built keeping the end application in mind, i.e., agriculture. Servo systems were used at each of the joints to allow precise movement and control over the arm. The servo motors are fitted with external 5:1 gears to improve the torque, thus improving their ability in loaded conditions. The servo motors at each of the joints are as given in Table 2:

Table 2. Servo motors at each of the joints

Joint	Specification
Shoulder	HS 805BB, 24. 7kg.cm Torque
Elbow	HS 805BB, 24. 7kg.cm Torque
Wrist	HS 785HB, 13. 2kg.cm Torque

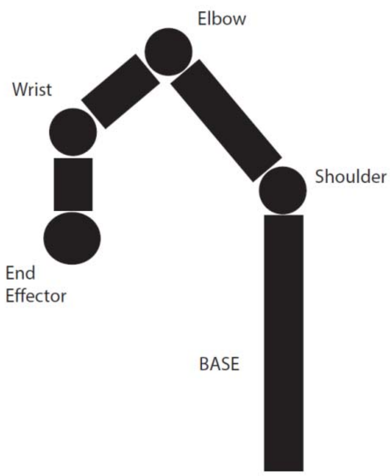

Figure 4. The diagram of the robotic arm showing each of the joints.

Figure 4 shows the basic outline of the robotic arm we made. Each joint of the robotic arm was appropriately mapped with the joint of the right human arm. The arm at the native reset position stays upright as depicted. This is done by keeping the end agricultural application in mind. The upright reset position of the arm will allow it to be more flexible in the real time application.

VI. Implementation

Implementation of the arm is divided into two parts, i.e., implementation of hardware and implementation of software. The hardware implementation includes all the components – Microsoft Kinect, aluminum links, servo motor systems, control system and power supply. Aluminum links are used to make a balance between material cost and weight to arrive at a design which is cost effective as well as light. The actuators used in the robotic arm are servo motors. The required torque at each of the joints is calculated and the corresponding servo motor system is chosen to meet the predicted requirements. The control system consists of a portable computer equipped with Kinect SDK, and an Arduino UNO. The sole purpose of Arduino UNO is to control servo links and give feedback to the Portable computer. The communication is via serial port. The power supply consists of 2 Li-Ion batteries, each with a capacity of 2200 mAh. Figure 5 shows the arm constructed. It uses servo motors as actuators at joints and the links are made of aluminum. The aluminum links are attached with the servo motors using 5:1 gears.

Figure 5. The robotic arm constructed using aluminum links, mounted in the laboratory.

A. Jacobian Iteration

From the point of robotics applications, several methods exist for solving IK problems. Cyclic coordinate descent methods, pseudoinverse methods, Jacobian transpose methods, Levenberg-Marquardt damped least squares methods and triangulation method are a few of the methods commonly used.

In our paper, we will be solving inverse kinematics problem using Jacobian iteration method

In robotics, the Jacobian iteration can be interpreted as the time derivative of kinematic equation which relates the velocity of the end effector to the joint rates. The motto of using this technique is to vary the joint orientations from a starting position (rest) to the desired end effector position. On each iteration there is small amount of change and this is defined by the relationship between the partial derivatives of joint angles θ, and difference between the current location of the end effector and desired position. This is a matrix that has dimensionality (p x q) where p is the dimension of end effector and q is the size of the joint orientation set.

Configuration of multi body is defined by scalars $\theta 1, \ldots \ldots \theta n$. These are the joint configurations. If a body has n joints and θj will denote to the joint angle. If there are k number of end effectors, then $s1....s_k$ are position of end effectors. Robotic arm will be controlled will be controlled by specifying the target positions for end effectors. The target position are also given by a vector

$t = (t1,...ti)^T$

Let ei = ti - si , the desired change in position of the ith end effector where

ti = target position for ith end effector

si = position of ith end effector.

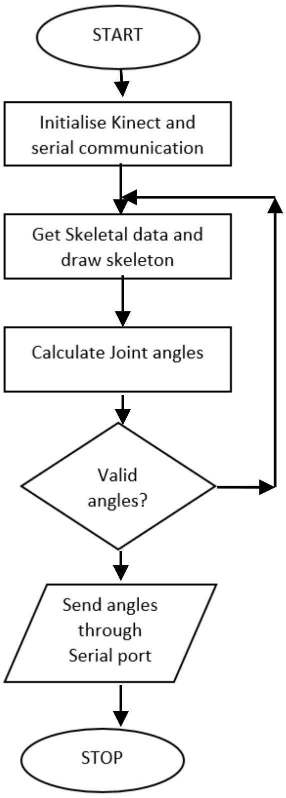

Fig 6: Kinect programming flow chart

Joint angles are written in column vector (θ1 .... θn)T End effector position can be also represented as function of joint angles given by

s = s(θ)

Inverse kinematics problem is to find values of θ_j. Equation 1 may not always give a solution and it may not be the best solution that is required

$$t_i = s_i(\theta) \text{ -------- } (1)$$

Therefore, we are going for iterative methods to get better solution for inverse kinematics problems. For this, the functions s_i are linearly approximated using the Jacobian matrix. The Jacobian Matrix depends on the joint angle and the position of the end effector The Jacobian Matrix J is a function of the θ values and is defined by

$$J\ \theta = \frac{\partial s}{\partial \theta}$$

J can be viewed as p × q matrix with scalar entries (with m = pk). In case forward dynamics, the equation which describes the velocities of the end effectors is given as

$$\partial s = J(\theta)\partial\theta$$

Equation (1) can be solved using iterative method From the given values of θ, s and t . From these Jacobian $J = J\ \theta$ is computed. We increment the value of θ to θ + Δθ, so new value of θ is given by

$$\theta = \theta + \Delta\theta \text{ -------- } (2)$$

Whenever a change in θ is made there will be always a change occurred in end effecter position given by

$$\Delta s \approx J(\Delta\theta)$$

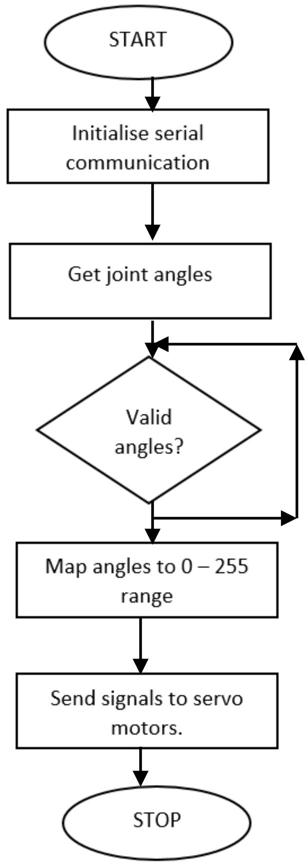

Fig 7:Arduino programming flow chart

The idea behind incrementing the value of θ to Δθ is to make Δs approximately equal to e. For this to happen correct value of Δθ must be chosen.

After each step is performed, the new the new joint angles make the end effector achieve the desired position approximately. This process keeps on going until value of s is sufficiently close to the solution. With more number of iterations, the end effector position is tracked more accurately.

VII. Experiment and Results

The Kinect can detect 2 people. To avoid clash, the code was written such that it recognizes only one person. The joint data was taken from the Kinect and it was processed using inverse kinematics. The program is made such that it operates only on

the arm joints from the entire body detected by it. Movement of the whole body would be shown but the hand part alone would be taken as input for calculations. The data from the Kinect is processed it using inverse kinematics algorithm and the calculated result is sent to the Arduino via serial port. Thus, the Arduino is able to control the robotic arm. The 3 DOF robotic arm was able to mimic the human arm movements at an acceptable rate, with a small delay.
The Kinect was tested and these are some of the results. As shown fig.8a, one hand was raised before the Kinect and the reading can be seen in fig.8b. And when the other hand is raised in fig.8c, the reading is correspondingly shown in fig.8d. As seen in these figures, there are three joints namely shoulder, elbow and arm. These are read using Kinect and can be processed accordingly.

To test the Kinect module, few experiments were conducted which helped in the analysis of this new system. A volunteer was asked to show different gestures and positions so that the ability of Kinect to detect these gestures is analyzed. There were six different positions which were experimented and analyzed but due to lack of space, only two figures are shown in figure 8. The two main angles which were given priority are the elbow angle (denoted by α) and the wrist angle (denoted by β) The readings for the same are given in the table 3.

Fig 8a. Kinect Testing

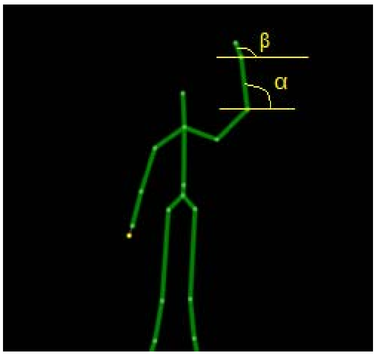

Fig. 8b. Kinect Capture Joints

Fig 8c. Kinect Testing with two arms

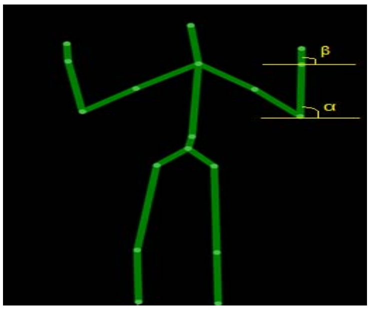

Fig 8d. Kinect Skeleton for two arms

Table 3. Positions of the Kinect simulation

Position	Elbow angle(α)	Wrist angle(β)
Position 1	101.25°	110°
Position 2	88.5°	89.75°
Position 3	274°	91°
Position 4	244°	210°
Position 5	320°	308°
Position 6	289°	288°

VIII. CONCLUSION

In this paper, we have presented solution to solve inverse kinematics problem using Jacobian Iteration method. We have worked on a way to interact with robotic arm using Kinect. We developed a robotic arm for this purpose and successfully controlled it with Kinect. As further scope of this method, we are looking into faster and more computationally efficient algorithms to solve for the inverse kinematics. We expect this to be of a greater scope in the future.

IX. ACKNOWLEDGEMENT

We wish to thank Almighty God who gave us the opportunity to successfully complete this venture. The authors wish to thank Amrita Vishwa Vidyapeetham University and the Humanitarian Technology (HuT) Labs for aiding us in this endeavor.

X. REFRENCES

[1] Hirose, S., Chu, R. “Development of a Lightweight Torque Limiting M-Drive Actuator for HyperRedundant Manipulator Float Arm”, ICRA 1999, pp.2831-2836

[2] Neo Ee Sian; Yokoi, K.; Kajita, S.; Kanehiro, F.; Tanie, K., "Whole body teleoperation of a humanoid robot - development

of a simple master device using joysticks," Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on , vol.3, no., pp.2569,2574 vol.3, 2002 doi: 10.1109/IRDS.2002.1041657

[3] Lim, S., Lee, K., & Kwon, D. (2003). Human friendly interfaces of robotic manipulator control system for handicapped persons. Proceedings of the 2003 IEEE/ASME …, (Aim), 435–440. Retrieved from http://robot.kaist.ac.kr/paper/article.php?n=53

[4] Assad, C., Wolf, M., Stoica, A., Theodoridis, T., & Glette, K. (2013). BioSleeve: A natural EMG-based interface for HRI. 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 69– 70. doi:10.1109/HRI.2013.6483505

[5] R. M¨uller-Cajar, R. Mukundan, 'Triangulation: A New Algorithm for Inverse Kinematics', Proceedings of Image and Vision Computing New Zealand 2007, pp. 181–186, Hamilton, New Zealand, December 2007

[6] K. C. Gupta and K. Kazerounian, "Improved numerical solutions of inverse kinematics of robots," in Proceedings of the 1985 IEEE International Conference on Robotics and Automation, vol. 2, March 1985, pp. 743–748.

[7] Pedro Neto, J. Norberto Pires & A. Paulo Moreira, Accelerometer-based control of an industrial robotic arm. Robot and Human Interactive Communication.2009, RO-MAN 2009

[8] Plateau, J. A. F. Statique expérimentale et théorique des liquides soumis aux seules forces moléculaires. *Gauthier-Villars*, 2 (1873).

[9] The New Industrial Revolution: 5 Technology Trends Driving Innovation, Forbes BrandVoice, Dec 10, 2015, www.forbes.com

[10] John Edwards, New Zealand Farming, Healthcare Robots Lead the Way to Commercialization, Sep 14, 2016, www.roboticsbusinessreview.com

[11] Tom Green, SRI Spins off Abundant Robotics & Vacuum Robot Harvester, Aug 10, 2016, www.roboticsbusinessreview.com

[12] http://groups.csail.mit.edu/drl/journal_club/papers/033005/buss-2004.pdf

[13] Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Ya el Eda N, "Vision-Based Hand-Gesture Applications," Communications of the ACM, vol . 54, pp.60-71, February 2011

[14] Darwin G.Caldwel1, O.Kocak and U.Andersen, "Multi-armed Dexterous Manipulator Operation using Glove/Exoskeleton Control and Sensory Feedback"

[15] Abidhusain Syed, Zamrrud Taj H. Agasbal, Thimmannagouday Melligeri , Bheemesh Gudur, "Flex Sensor Based Robotic Arm Controller Using Micro Controller", Journal of Software Engineering and Applications, 2012, pp. 364-366

[16] Rajesh Kannan Megalingam; Nihil Saboo; Nitin Ajithkumar; Sreeram Unny;Deepansh Menon

[17] Rajesh Kannan Megalingam; Nihil Saboo; Nitin Ajithkumar; Sreeram Unny;Deepansh Menon, "Kinect based gesture controlled Robotic arm: A research work at HuT Labs", 2013 IEEE International Conference in MOOC, Innovation and Technology in Education (MITE) Year: 2013 Pages: 294 - 299, DOI: 10.1109/MITE.2013.6756353